Plaintiffs' Exhibit 91

Investigative Memorandum: App Technical Investigation

TO:	Gibson Dunn & Crutcher LLP
FROM:	FTI Consulting
DATA AS OF:	July 11, 2018
REPORT AS OF:	August 13, 2018
RE:	Sync.Me (146237828595) Sync.Me (145845065580773) Sync.Me (565561836797777)
ESCALATED APP:	Sync.Me (565561836797777)
RECOMMENDATION:	Additional RFI / Potential Onsite

KEY POINTS

- **Summary App Description:** Sync.Me (the "App") was intended to act as a universal caller-id that would combine a user's contacts with social media to pull additional information and create one consolidated contact list [Source 1].
- **Reason for Secondary Review:** ALT
- **Level of concern: High**
    - The App held numerous sensitive permissions, most of which do not align with the understood use case.
    - The App heavily utilized methods that do not provide visibility into the endpoints accessed, but the ADI team believes it is likely the App accessed large amounts of the data governed by these permissions because of the following:
        - Previous enforcement history indicating the App was disabled for using these methods to scrape email addresses
        - Observable calls to data governed by sensitive permissions, to return posts and/or newsfeed data

- The App exhibits increased data request activity from late 2014 through mid-2015, which coincides with:
  - Teddy Sagi's acquisition of VisualDNA, an analytics firm that leverages personality quiz data for marketing. Mr. Sagi was an early investor in the App,
  - Transition from Graph API version 1.0 to 2.0

- **Summary of Findings:**

The Background Investigation [Source 2] found that the App has been flagged by numerous media outlets for storing contact information that is vulnerable to leaks and hacks and may be in violation of data protection rules [Source 18, Source 19]. Historic versions of the privacy policy confirm that names and phone numbers of users and their contacts are stored, but there is no discussion in the privacy policy regarding storage of other user data points believed to have been accessed by the App during Graph API v1.0 ("v1.0") [Source 22]. The App was part of an incubator run by Teddy Sagi and received an early investment from Sagi, who later purchased and then resold a company (Visual DNA) that leverages personality quiz data for marketing purposes [Source 20]. Sagi was also imprisoned for market manipulation prior to his involvement with Sync.me [Source 17]. The App's co-founder and former CEO (Schlomo Unger) was the founder of EZTrader, which has been involved in controversy surrounding trading practices and was fined by the SEC for illegally soliciting U.S. customers [Source 21]. Historic and current versions of the privacy policy indicate that data (including personal data) may be shared with affiliates, parent companies, and subsidiaries, or in the event of a corporate transaction [Source 22].

During v1.0, the App was over-permissioned and had access to many "heavyweight" permissions [Source 3]. Across all three versions, much of the App's request activity heavily utilized a method that provides no visibility into the data requested or returned. This method could have been used to access any data for which the App was permissioned. That same method was flagged by the Facebook enforcement team in 2013 for being leveraged to scrape data. The App could have been collecting a large amount of sensitive data on both the users and their friends. Further, in v2.0 – v3.0 ("v2.0"), App ID 565561836797777 utilized a method that Facebook described as a potential workaround for accessing friend's information despite a reduced set of permissions.

Exhibit Truncated Due to Size

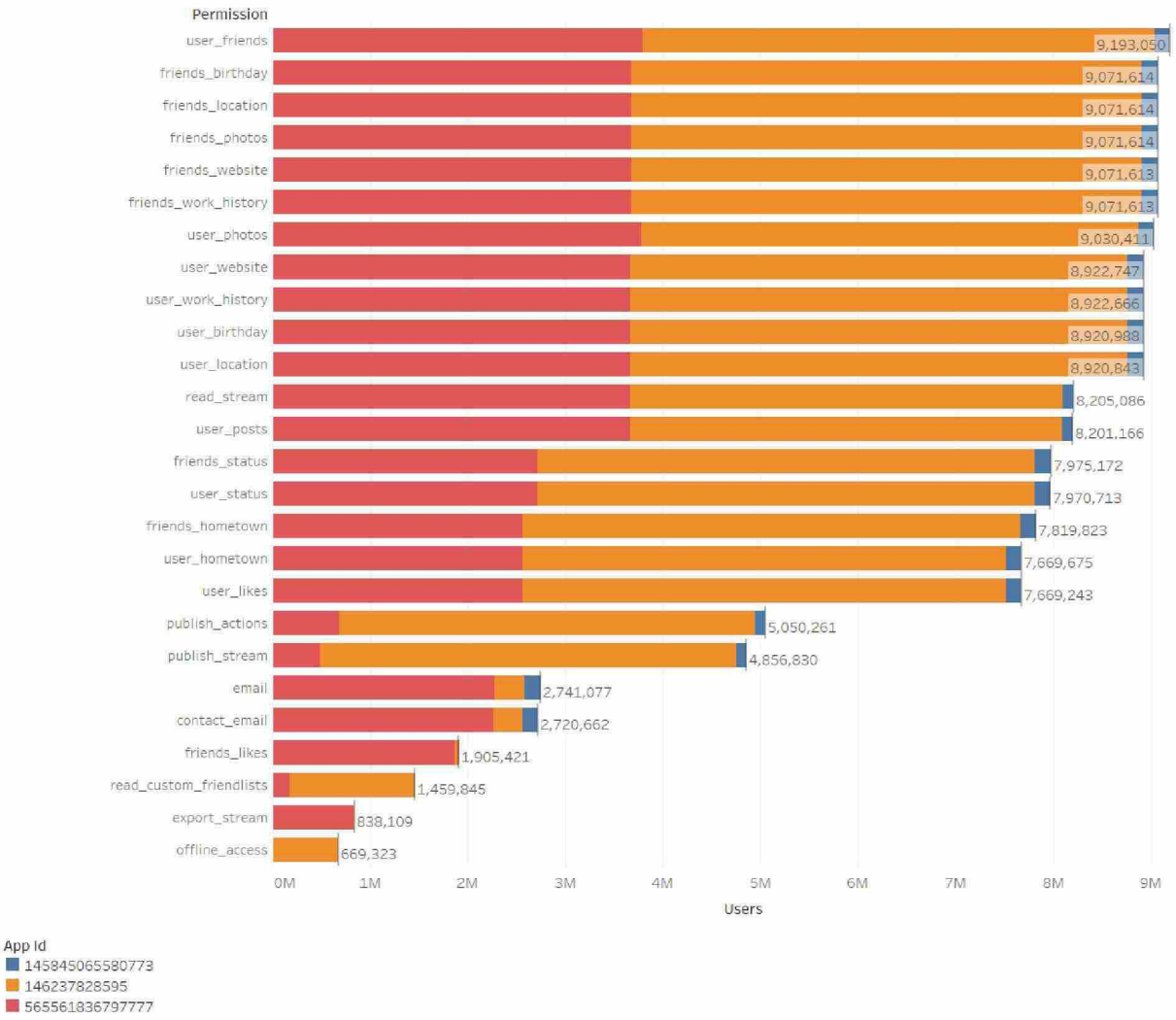

Figure 4: v1.0 Top Permissions Granted (by number of users)

Figure 4 reflects the top permissions granted (by number of users) to the App. All permissions may be reviewed in the supporting documentation [Source 12, 13, 14].

The App held a mixture of write and read permissions. For the purposes of this analysis, the ADI Team has not analyzed write permissions, which include any permissions with publish, create, upload, or share in the name as these permissions only allow the App to create posts on behalf of the user and do not provide access to user data. Instead, the ADI Team focused on read permissions, which allow the App to access information governed by that permission.

Based on information available to date and the understood use case of the App, it appears the App requested more permissions than necessary prior to the switch to v2.0, including some "heavyweight" permissions, which Facebook considers to be highly sensitive [Source 3].

- In aggregate for all three versions of the App, 8.9M users granted the App permission to read their location with user_location, and 9M users granted the App permission to read

friends_location. Note: the ADI team cannot determine if users overlapped between versions of the App, so this figure could be overstated.

- The App was granted access to permissions that provide information about user and friend likes (friend_likes was not requested for App ID 146237828595 only), user and friend's hometown, and user and friend's status.
- Read_stream and export_stream allow the App to access the user's entire newsfeed.
- User_posts allows the App to access all posts by a user.

These permissions grant the App access to sensitive information about users and their friends and appear to be out of scope for the use case of the App.

Exhibit Truncated Due to Size

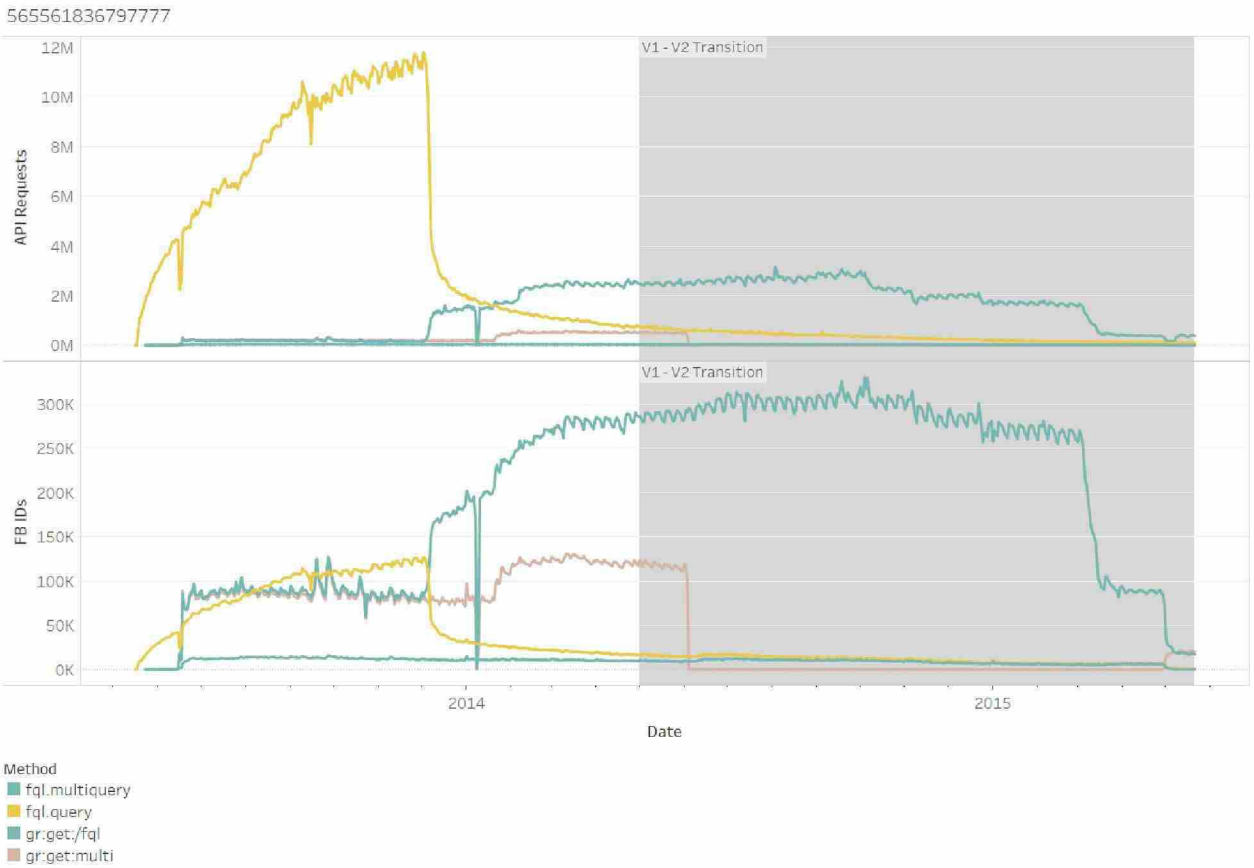

Figure 11: v1.0 Bulk Method Requests Over Time for App ID 565561836797777

A review of the methods called during the v1.0 period illustrates that a significant number of API Requests were made for the fql.query method.

- API requests made for fql.query represented 60% of the total API requests made by App ID 146237828595.
- For App IDs 145845065580773 and 565561836797777, the FQL method is nearly 23% of all calls.

Based on data available at this time, the methods logged as fql.query and fql.multiquery appear to be fql calls to the legacy REST API, and the methods logged with gr:get are fql calls to the Graph API.

These facts, paired with the high-risk permissions the App was granted during v1.0, make it possible that the App was accessing sensitive user data that did not align with its use case.

This conclusion is further corroborated by:

1. Enforcement notes provided by the ALT review for App ID 565561836797777which include: "Very sketchy enforcement taken on the app in 2013, prior to locking down the platform. Excerpts of internal policy notes: Original enforcement reason: 'Making tons

user searches for email address via API, has 2 previous apps disabled for same service' later: 'Request from Shirine to re-enable the app. **Cullen saw a bunch of fql queries searching for email addresses, and disabled them for being a scraping app**. Shirine requested that we restore it as we no longer disable apps like these for violating what at the time was FPP I.12.'"

a. The ADI team bolded the section of the above quote which indicates the App had been enforced upon in the past for scraping data using FQL queries.

2. Known calls to the user/feed endpoint, which returns sensitive data (given the permissions profile), detailed below.

Figure 12: v1.0 API Requests to User/feed over time

Though used infrequently in comparison with the Bulk methods highlighted above, Figure 12 is an example of the App leveraging the user_posts and read_stream permissions and was observed across all three versions of the App. Given the access to those permissions, gr:get:User/feed would have returned at least all posts made by a user but may have also returned posts by others on that user's feed. App ID 565561836797777 also exhibits a spike in activity for gr:get:User/feed during the v1.0-v2.0 transition period.